***NOLLE***

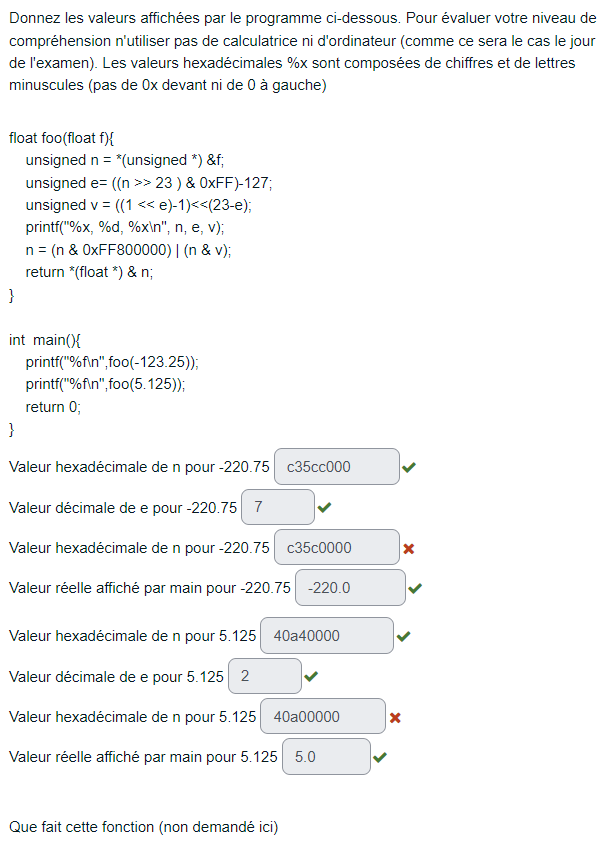
***Damien***

***L3 – Informatique***

**ADO – Devoir 2 :**

|  |  |
| --- | --- |
| ***Note :*** | ***Observation :*** |
| ***/20*** |  |

Exercice 1)



La première instruction permet de récupérer la représentation IEEE 754 (la représentation mémoire) d’un réel simple précision (float).

-220.75

220 = 1101 1100

220 – 128 = 92 – 64 = 28 – 16 = 12 – 8 = 4 – 4 = 0

0,75 = 11

0.5 + 0.25 = 0.75

220,75 = 1101 1100,11

Représentation scientifique :

1,101110011 \* 27

Mantisse : 101110011

7 + 127 = 134

134 – 128 = 6 – 4 = 2 – 2 = 0

1000 0110

Puisque le nombre est négatif, le bit de signe est à 1.

IEEE 754 :

1100 0011 0101 1100 1100 0000 0000 0000 = 0xC35CC000

8 + 4 = 12 🡪 C

1 + 2 = 3

4 + 1 = 5

8 + 4 = 12 🡪 C

8 + 4 = 12 🡪 C

0

0

0

e = ((n >> 23) & 0xFF) – 127

1100 0011 0101 1100 1100 0000 0000 0000 >> 23 = 0000 0000 0000 0000 0000 0001 1000 0110

0000 0000 0000 0000 0000 0001 1000 0110 & 0xFF

0000 0000 0000 0000 0000 0001 1000 0110

& 0000 0000 0000 0000 0000 0000 1111 1111

0000 0000 0000 0000 0000 0000 1000 0110

1000 0110 = 128 + 4 + 2 = 134

134 – 127 = 7

1100 0011 0101 1100 1100 0000 0000 0000 & 0xFF800000

1100 0011 0101 1100 1100 0000 0000 0000

& 1111 1111 1000 0000 0000 0000 0000 0000

1100 0011 0000 0000 0000 0000 0000 0000

v = ((1 << e) – 1) << 23 – e

(1 << 7) - 1 = 1000 0000 – 1 = 1000 0000 + (-1)

NOT(1) + 1 = NOT(0000 0001) + 1 = 1111 1110 + 1 = 1111 1111

1000 0000 + 1111 1111 = 0111 1111

0111 1111<< 23 – e = 0111 1111<< 23 – 7 = 0111 1111 << 16 = 0111 1111 0000 0000 0000 0000

1100 0011 0101 1100 1100 0000 0000 0000 & 0111 1111 0000 0000 0000 0000

1100 0011 0101 1100 1100 0000 0000 0000

& 0000 0000 0111 1111 0000 0000 0000 0000

0000 0000 0101 1100 0000 0000 0000 0000

1100 0011 0000 0000 0000 0000 0000 0000

| 0000 0000 0101 1100 0000 0000 0000 0000

1100 0011 0101 1100 0000 0000 0000 0000

C35C0000

1000 0110 = 128 + 4 + 2 = 134 – 127 = 7

1,10111 \* 27 = 1101 1100 = 128 + 64 + 16 + 8 + 4 = 192 + 28 = -220

5.125

5 = 101

0,125 = 001

101,001

1,01001 \* 22

2 + 127 = 129 = 1000 0001

0100 0000 1010 0100 0000 0000 0000 0000 = 0x40A40000

4

0

8 + 2 = 10 🡪 A

4

0

0

0

0

e = ((0100 0000 1010 0100 0000 0000 0000 0000 >> 23) & 0xFF) – 127

0000 0000 0000 0000 0000 0000 1000 0001

& 0000 0000 0000 0000 0000 0000 1111 1111

0000 0000 0000 0000 0000 0000 1000 0001

129 – 127 = 2

v = ((1 << 2) – 1) << (23 – 2)

100 – 1 = 100 + (-1)

Not(1) + 1 = 110 + 1 = 111

100 + 111 = 011

011 << 21 = 0110 0000 0000 0000 0000 0000

n = (0100 0000 1010 0100 0000 0000 0000 0000 & 0xFF800000) | (0100 0000 1010 0100 0000 0000 0000 0000 & 0110 0000 0000 0000 0000 0000)

0100 0000 1010 0100 0000 0000 0000 0000

& 1111 1111 1000 0000 0000 0000 0000 0000

0100 0000 1000 0000 0000 0000 0000 0000

0100 0000 1010 0100 0000 0000 0000 0000

& 0000 0000 0110 0000 0000 0000 0000 0000

0000 0000 0010 0000 0000 0000 0000 0000

0000 0000 0010 0000 0000 0000 0000 0000

| 0100 0000 1000 0000 0000 0000 0000 0000

0100 0000 1010 0000 0000 0000 0000 0000

129 – 127 = 2

1,01 \* 22

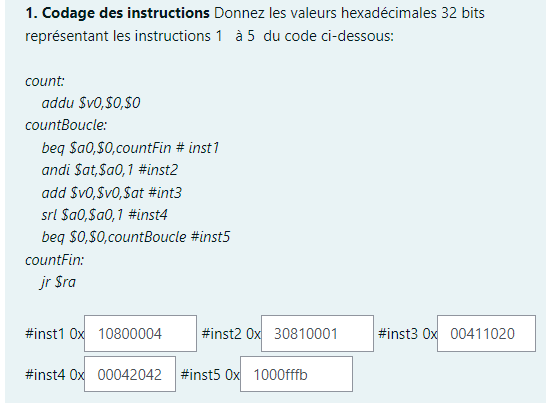
101 = 4 + 1 = 5

0100 0000 1010 0000 0000 0000 0000 0000 = 0x40A0 0000

Cette fonction permet de calculer la partie entière d’un nombre réel simple précision (float), à partir du nombre au format IEEE 754.

Exercice 2)

Q1 :



beq $a0, $0, countFin

C0 = 4 = 0001 00

rs = $a0 🡪 $4, 4 = 0 0100 (5bits)

rt = $0, 0 = 0 0000

countFin : 🡪 4 = 0100

(cp + 4) + (immS16 SLL 2)

4 + 4 = 8

4 SLL 2 = 0001 0000 = 16

16 + 8 = 24

0001 0000 1000 0000 0000 0000 0000 0100

0x10800004

andi $at, $a0, 1

C0 = 0xC = 00 1100

rt 🡪 $at 🡪 $1 = 0 0001

rs 🡪 $a0 🡪 $4, 4 = 0 0100

immNs16 🡪 1 = 0000 0000 0000 0001

0011 0000 1000 0001 0000 0000 0000 0001

0x30810001

add $v0, $v0, $at

C0 = 00 0000

rd 🡪 $v0 🡪 $2 = 0 0010

rs 🡪 $v0 🡪 $2 = 0 0010

rt 🡪 $at 🡪 $1 = 0 0001

shamt = 0 0000

Nf = 10 0000

0000 0000 0100 0001 0001 0000 0010 0000

0x00411020

srl $a0, $a0, 1

C0 = 00 0000

rd 🡪$a0 🡪 $4, 4 = 0 0100

rt 🡪 $a0 🡪 $4, 4 = 0 0100

shamt 🡪 0 0001

Nf = 00 0010

0000 0000 0000 0100 0010 0000 0100 0010

0x00042042

beq $0 $0, countBoucle

Co = 4 = 0001 00

rs 🡨 0 0000

rt 🡨 0 0000

ImmS16 = -5 = 1111 1111 1111 1011

0001 0000 0000 0000 1111 1111 1111 1011

0x1000FFFB

8+2+1 = 11 🡪 B

4/4 – (20 + 4) / 4 = 1 – 6 = -5

(Cp + 4) + (ImmS16 SLL 2)

20 + 4 = 24

5 = 0000 0000 0000 0101

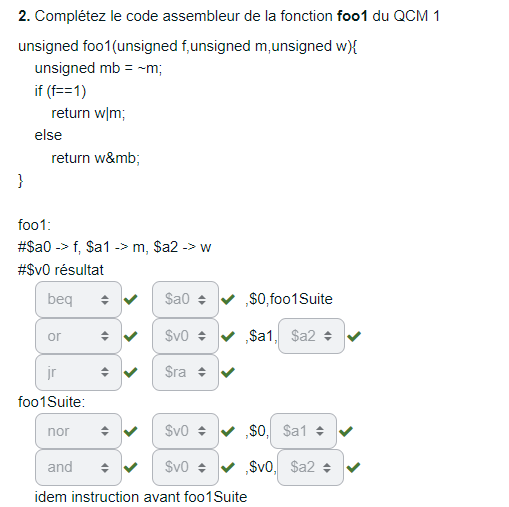
NOT(0000 0000 0000 0101) + 1

1111 1111 1111 1010 + 1 = 1111 1111 1111 1011

1111 1111 1111 1011 SLL 2 = 1111 1111 1110 1100

24 + -20 = 24 – 20 = 4

Q2:

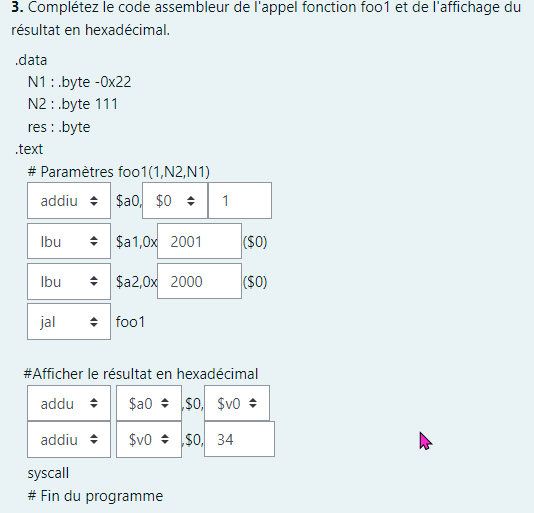


4 = 0100

0 = 0000

nor 1011

Q3 :



Exercice 3)

